Extraction d'information à base d'ontologies dans une application de veille*

A.Todirascu (1), L.Romary (2), D.Bekhouche (3)

(1) Université de Technologie de Troyes, 12, rue Marie Curie 10010 Troyes Cedex
(2) Laboratoire Loria - INRIA Lorraine
Campus scientifique, BP 239, 54506 Vandoeuvre-les-Nancy Cedex
(3) EADS/Systèmes & Défenses Electroniques, Louvier

Résumé

Le papier présente un système d'extraction d'information, qui s'appuie sur des techniques robustes d'analyse du langage naturel et sur l'existence d'une ontologie d'un domaine. Nous présentons en particuliér l'interface syntaxe-sémantique mise en oeuvre pour un corpus des messages électroniques (en anglais) sur la sécurité des systèmes informatiques.

This paper presents an information extraction system dedicated to message filtering for a specific domain. We focus on a method for identifying domain-specific entities, using syntactic information and an existing domain ontology. The application domain concerns security in computer systems and an English corpus of electronic messages has been used for tests.

1. Introduction

Le but des systèmes d'extraction d'information est d'identifier les entités pertinentes dans les textes (pour un domaine limité ou générique) à l'aide de bases de connaissances du domaine. Ces systèmes sont appliqués sur des textes de grande taille, qui peuvent contenir des erreurs, il est donc nécessaire de s'appuyer sur des techniques d'analyse de surface du langage naturel (Daille, 1996). Les bases de connaissances génériques sont peu adaptées aux applications sur un domaine limité et la construction d'une base spécifique exhaustive et libre d'incohérences est très coûteuse et très peu portable.

Pour certains scénarios spécifiques, il est cependant possible de mettre en oeuvre une méthodologie combinant analyse linguistique et une ontologie de référence, validée par des experts. Dans le cadre du projet Vulcain, et en collaboration avec EADS/SD&E, nous avons ainsi pu travailler à la définition d'un système de suivi d'événements relatifs à la sécurité informatique, évènements s'articulant autour d'un certain nombre de concepts de base identifiés par les experts du domaine (e.g. accès, intrusion, hacker, contournement, etc.). A partir de ceux-ci, nous

*Ce travail a été effectué dans le cadre d'un projet (Vulcain) soutenu par la DGA et en collaboration avec EADS/SD&E

avons identifié des comportements syntaxiques locaux associés à ces concepts, nous permettant d'envisager leur détection automatique et leur combinaison éventuelle au sein d'une même unité linguistique (syntagme, phrase ou document).

Nous avons choisi d'utiliser une ontologie décrite dans un formalisme de représentation des connaissances qui permet d'utiliser des inférences pour valider les candidats identifés par des analyses syntaxiques partielles. Nous nous focalisons sur la liaison entre les ressources syntaxiques et l'ontologie.

2. L'architecture

Nous proposons une architecture (cf. fig. 1) qui utilise un corpus de référence pour créer des ressources spécifiques au domaine. Le corpus est utilisé pour créer un lexique et une grammaire spécifique, à partir d'une grammaire et d'un lexique TAG (Joshi, 1987) existantes. Le même corpus est utilisé pour créer l'ontologie de base, qui valide les candidats identifiés dans le texte par l'analyse syntaxique partielle.

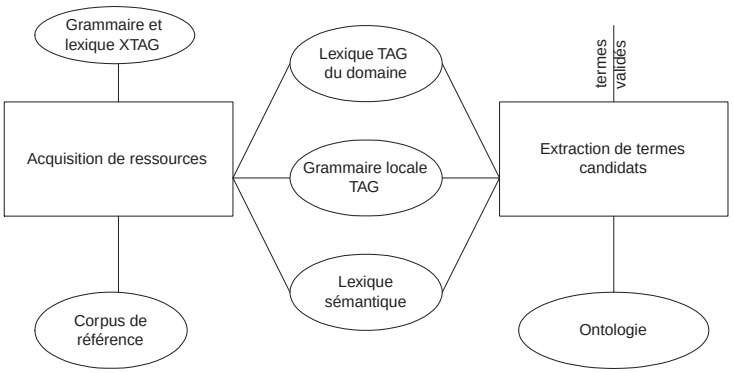

Figure 1: L'architecture du système

3. Les ontologies

Pour notre application, l'ontologie est un modèle simplifié des connaissances du domaine, représentées comme des concepts reliés par des relations. Pour construire une telle ontologie, plusieurs outils ont été développés pour extraire les candidats termes (Daille, 1996) et pour identifier des relations entre les classes des termes (Cerbah, 2000). Un expert humain associe une interprétation aux concepts, et aux relations entre les concepts (Biebow & Szulman, 2000). Nous avons adopté une approche logique qui essaie de résoudre les problèmes de manque de corpus d'entrainement requis par les méthodes statistiques.

Le rôle de l'ontologie dans notre système est de valider les entités identifiées dans le texte par les techniques du traitement automatique du langage naturel. On doit pouvoir exploiter les relations hiérarchiques entre les concepts ainsi que des connaissances implicites. Nous avons ainsi choisi un formalisme de représentation de connaissances qui permette de faire des inférences (les logiques de description).

Les logiques de description regroupent plusieurs propriétés des réseaux sémantiques, des systèmes orientés objet, des systèmes des frames, ayant une syntaxe et une sémantique bien

définis (Brachman & Schmolze, 1985). Les ontologies représentées dans une logique de description sont utiles pour valider des résultats de l'analyse syntaxique, à l'aide d'algorithmes qui calculent d'une manière décidable les relations entre les concepts du domaine et les liaisons entre les concepts et leurs instances. Une autre propriété intéressante est la manipulation des données semi-structurées ou incomplètes. Par exemple, pour l'instance x, il n'est pas nécessaire de définir explicitement les valeurs des rôles **hasOSParent, hasSize, hasName** :

```
(define-concept file (and physical-object (all hasOSParent OSystem)(all hasSize
Size)(all hasName Name)(some hasType Type)))

(instance x (and file (some hasType binary)))
```

Pour notre système, nous avons besoin de définir des concepts, de calculer des chemins de rôles entre deux concepts, de raisonner au niveau des instances. Parmi les systèmes de LD, nous avons choisi RACER (Haarslev & Muller, 2001) qui propose la plupart de ces fonctionalités.

L'ontologie a été construite manuellement à partir d'un corpus de référence de 50000 occurences (4039 mots). Nous avons extrait une liste un certain nombre de verbes et de noms parmi les plus fréquents. A partir de cette liste, nous avons défini 38 concepts et 41 rôles entre concepts, et nous avons utilisé RACER pour vérifier la cohérence de cette ontologie.

4. Les ressources linguistiques

Nous présentons les ressources linguistiques nécessaires pour extraire les informations. Nous avons choisi comme analyseur syntaxique le parseur TAG de Patrice Lopez (Lopez, 1999), qui utilise des grammaires d'adjonction d'arbres (Joshi, 1987) et qui fournit des analyses partielles.

Le corpus de référence est formé par un ensemble des messages électroniques sur la sécurité des systèmes informatiques. Il contient beaucoup d'erreurs de syntaxe et d'orthographe, des noms de fonctions ou des variables, des commandes Unix ou Dos, des syntagmes qui introduisent un dialogue, qui font necessaire l'utilisation des méthodes robustes d'analyse. **Le lexique** XTAG de test pour l'anglais, qui contient 500 entrées (Barthélemy *et al.*, 2001) a été utilisé pour créer un lexique spécifique du domaine, à l'aide du catégoriseur lexical TreeTagger (Schimd, 1994) et du corpus. Le lexique contient 2500 noms, 50 verbes, 767 adjectifs, 379 adverbes et 105 prépositions. **La grammaire** de l'anglais XTAG, qui contient 421 arbres élémentaires, est utilisée pour créer une sous-grammaire, qui contient les arbres élémentaires pour les groupes nominaux et prépositionnels, mais aussi des verbes du domaine.

L'interface syntaxe-sémantique Le résultat d'un analyseur TAG est un ensemble d'arbres de dérivation et d'arbres dérivés. Nous proposons une méthode qui transforme les arbres de dérivation dans une description conceptuelle, qui sera validée par l'ontologie du domaine. Pour cela, nous avons besoin d'un lexique sémantique qui fournit une liste de correspondances entre les lemmes et les descriptions conceptuelles. Les arbres de dérivation montrent comment ont été combinés les arbres élémentaires pendant l'analyse syntaxique.

Exemple. Pour la phrase "The root downloads the file from the server", les concepts sont donnés dans le tableau 1. Nous avons appliqué les contraintes de *download* pour obtenir deux concepts complexes, à valider par l'ontologie :

```
(and download (some agent (and root (some hasType Def))) (some patient (and file
(some hasType Def))) (and from (some hasplace (and server (some hasType Def)))))
```

Mots	Concepts
downloads	(and download (some hasSubstitution (and x (some address 1)))(some hasSubstitution (and y (some address 3)))) (implies (some hasSubstitution x) (some agent x)) (implies (some hasSubstitution y)(some patient y))
root, server, file	root, server, file
the	(some hasType Def)
from	(and from (some hasplace concept))

Table 1: Les concepts extraits de la phrase "The root downloads the file from the server"

```
(and download (some agent (and root (some hasType Def))) (some patient (and file
(some hasType Def)(and from (some hasplace (and server (some hasType Def )))))))
```

5. Conclusion et perspectives

L'article presente les modalités de combiner la grammaire LTAG avec une ontologie du domaine pour identifier les concepts possibles. A partir du travail que nous avons realisé en grande partie manuellement, nous allons proposer des extensions automatiques des ressources (les arbres elementaires et les descriptions conceptuelles associées), à l'aide d'une meta-grammaire. Pour automatiser la tâche de mise à jour des ressources linguistiques, nous développons une méthode pour generer les entrées lexicales des verbes d'une manière automatique.

Références

BARTHÉLEMY F., P. B., P. D., KAOUANE L., KHAJOUR A. & VILLEMONTE DE LA CLERGERIE E. (2001). Tools and resources for tree adjoining grammars. In *Proceedings of ACL'01 workshop on Sharing Tools and Resources*, p. 63–70, Toulouse, France.

BIEBOW B. & SZULMAN S. (2000). Une approche terminologique pour catégoriser les concepts d'une ontologie. In J. CHARLET, M. ZACKLAD, G. KASSEL & D. BOURIGAULT, Eds., *Ingénierie des connaissances - Evolutions récentes et nouveaux défis*, p. 325–336: Eyrolles Publishing House.

BRACHMAN R. & SCHMOLZE J. (1985). An overview of the kl-one knowledge representation system. volume 9, p. 171–216.

CERBAH F. (2000). Exogeneous and endogeneous approaches to semantic categorization of unknown technical terms. In *Proceedings of COLING'2000*, p. 145–151, Nancy - Luxembourg - Saarbrucken.

DAILLE B. (1996). Study and implementation of combined techniques for automatic extraction of terminology. In J. KLAVANS & P. RESNIK, Eds., *The Balancing Act - Combining Symbolic and Statistical Approaches to Language*, p. 49–66: MIT Press.

HAARSLEV V. & MULLER R. (2001). Description of the racer system and its applications. In *International Workshop on Description Logics (DL-2001)*, p. 132–141, Stanford.

JOSHI A. (1987). An introduction to tree adjoining grammars. *Mathematics of Language*, **1**, 87–115.

LOPEZ P. (1999). *Analyse syntaxique robuste avec les grammaires d'arbres adjoints lexicalisées pour les systèmes de dialogue*. Nancy: PhD. Thesis, INRIA.

SCHIMD H. (1994). Probabilistic part-of-speech tagging using decision trees. In M. PAZIENZA, Ed., *International Conference on New Methods in Language Processing*, Manchester, United Kingdom: Springer-Verlag.